Chapter 3
Operational Requests

This chapter explains how to use the JUNOScript application programming interface (API) to request information about router status. The JUNOScript request tag elements described here correspond to JUNOS command-line interface (CLI) operational commands described in the JUNOS Internet software operational mode command references. The JUNOScript API defines a specific request tag element for most commands in the CLI show family of commands. For example, the <get-interface-information> tag element corresponds to the show interfaces command, and the <get-chassis-inventory> tag element requests the same information as the show chassis hardware command.

This chapter discusses the following topics:

For information about using the JUNOScript API to request router configuration information, see “Router Configuration” on page 41.

Request Operational Information

To request operational information from the JUNOScript server, first establish a connection to it and open a JUNOScript session, as described in “Connect to the JUNOScript Server” on page 18 and “Start the JUNOScript Session” on page 19.

Then emit operational request tag elements, each enclosed in an opening <rpc> and closing </rpc> tag. Client applications can emit an unlimited number of operational request tag elements during a JUNOScript session (one at a time). It can freely intermingle them with configuration requests, which are described in “Router Configuration” on page 41.

Each operational request tag element corresponds to a JUNOS CLI command that has a distinct function or returns a different kind of output. For a list of mappings between CLI commands and operational request tag elements, see the *JUNOScript API Reference*.

The JUNOScript API defines separate Extensible Markup Language (XML) document type definition (DTD) files for different JUNOS components. For instance, the DTD for interface information is called junos-interface.dtd and the DTD for chassis information is called junos-chassis.dtd. Each DTD constitutes a separate XML namespace, which means that multiple DTDs can define a tag element with the same name but a distinct function. The *JUNOScript API Reference* includes the text of the JUNOScript DTDs.

Client applications can emit all tag elements that constitute a request by invoking one instance of a routine such as write( ), or can invoke a separate instance of the routine for each tag element or group of tag elements. The JUNOScript server ignores any newline characters, spaces, or other white space characters in the tag stream.

After the client application finishes making requests, it can close the JUNOScript session and terminate the connection, as described in “End the Session and Close the Connection” on page 31.

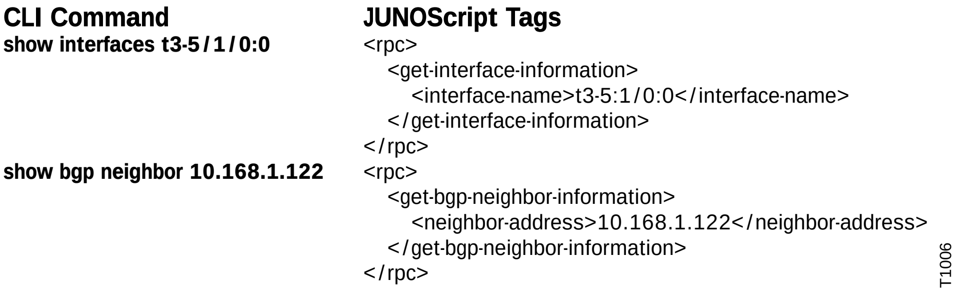
JUNOS CLI commands take two main kinds of options, and there are matching child tag elements in the corresponding operational request tag element. For a discussion and example of each, see the following sections:

- Map Child Tag Elements to Options with Variable Values on page 38
- Map Child Tag Elements to Fixed-Form Options on page 39

Map Child Tag Elements to Options with Variable Values

Many JUNOS CLI commands have options that specify the name of the object that the command affects or reports about. In some cases, the CLI does not precede the option name with a fixed-form keyword, but XML convention requires that the JUNOScript API define a tag element for each option. The tag element is most often called <name>, but other names are sometimes used. To learn the names for the child tag elements of an operational request tag element, consult the tag element’s entry in the appropriate DTD or in the *JUNOScript API Reference*.

The following illustrates the mapping between CLI command and JUNOScript tag elements for two CLI operational commands with variable-form options. In the show interfaces command, t3-5/1/0:0 is the name of the interface. In the show bgp neighbor command, 10.168.1.222 is the IP address for the BGP peer of interest.



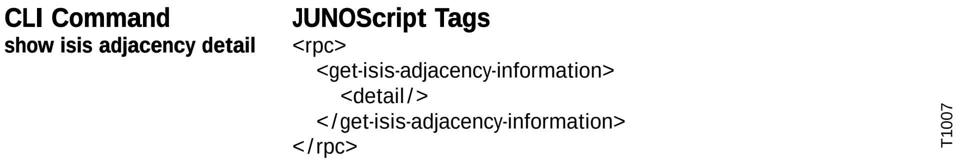
```
CLI Command                        JUNOScript Tags
show interfaces t3-5/1/0:0         <rpc>
                                     <get-interface-information>
                                       <interface-name>t3-5:1/0:0</interface-name>
                                     </get-interface-information>
                                   </rpc>
show bgp neighbor 10.168.1.122     <rpc>
                                     <get-bgp-neighbor-information>
                                       <neighbor-address>10.168.1.122</neighbor-address>
                                     </get-bgp-neighbor-information>
                                   </rpc>
```

T1006

Map Child Tag Elements to Fixed-Form Options

Some JUNOS CLI commands include options that have a fixed form, such as the brief and detail strings, which specify the amount of detail to include in the output. The JUNOScript API usually maps such an option to an empty tag element whose name matches the option name.

The following illustrates the mapping between the CLI command and JUNOScript tag elements for the show isis adjacency command, which has a fixed-form option called detail:



CLI Command

show isis adjacency detail

JUNOScript Tags

```
<rpc>
  <get-isis-adjacency-information>
    <detail/>
  </get-isis-adjacency-information>
</rpc>
```

T1007

Parse an Operational Response

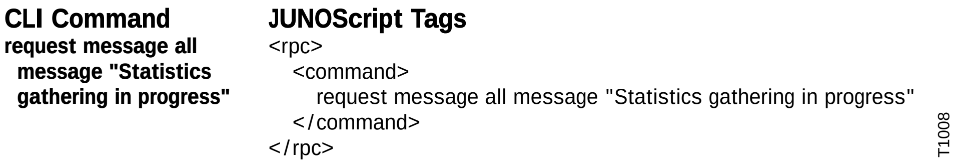
The JUNOScript server encloses its response to each operational request in a separate <rpc-reply> tag element. For information about the conventions that client applications must follow when interpreting a JUNOScript server response, see “General JUNOScript Conventions” on page 9. For information about parsing a response tag element, see “Parse the JUNOScript Server Response” on page 28.

Client applications must include code for accepting the tag stream and extracting the content from tag elements. They can, for instance, feed them to a parser that implements a standard API such as the Document Object Model (DOM) or Simple API for XML (SAX). For more information, see “Use a Standard API to Parse Response Tag Elements” on page 31.

Requests and Responses without Defined JUNOScript Tag Elements

The JUNOScript API does not define a tag element for every JUNOS CLI command. Client applications can still invoke the functionality of such commands by enclosing the actual CLI command string in a <command> tag element. Like all request tag elements, the <command> tag element must occur within an <rpc> tag element.

The following illustrates the use of the <command> tag element to issue the request message command, for which the JUNOScript API does not define a tag element. The line breaks in both the CLI command and within the <command> tag elements are for legibility only. The client application should not include newlines or other extraneous white space characters in the string within the <command> tag elements. Also, unlike most examples in this manual, the following does not preserve a line-to-line correspondence between the CLI command and the JUNOScript tag elements:



CLI Command

```
request message all
  message "Statistics
  gathering in progress"
```

JUNOScript Tags

```
<rpc>
  <command>
    request message all message "Statistics gathering in progress"
  </command>
</rpc>
```

T1008

If the JUNOScript API does not define a response tag element for the type of output requested by a client application, the JUNOScript server encloses its response in an <output> tag element. The tag element's contents are usually one or more lines of formatted ASCII output like that displayed by the JUNOS CLI on the computer screen.

> Caution
>
> The JUNOScript server might not support use of all CLI commands in the <command> tag element. For details, see the JUNOS Internet software release notes.
>
> The content and formatting of data within an <output> tag element are subject to change, so client applications must not depend on them. Future versions of the JUNOScript API will define specific response tag elements (rather than <output> tag elements) for more commands. Client applications that rely on the content of <output> tag elements will not be able to interpret the output from future versions of the JUNOScript server.